算法分析

定义及目的

- 算法分析：估计算法需要资源的程度。
- 目的1：选择算法
  - 对同一个问题可以设计不同的算法，它们有不同的时间和空间需求；
  - 分析算法可以了解算法的性能，通过比较不同算法的性能，可以选择好的算法避免人力和物力浪费。
- 目的2：优化算法
  - 分析算法可以更加深入了解算法，发现其中的缺陷与不足，从而可以进一步优化算法。

计算模型

- **单处理器计算模型：随机访问机(random-access machine: RAM)**
  - 指令顺序执行，没有并发操作；
  - 包含常用指令，每条指令执行时间为常量；
  - 数据包含整数类型和浮点实数类型；
  - 不对存储器层次进行建模。

- **Note：默认情况下，算法分析一般是指分析算法的时间效率。**

时间分析

- 相关因素
  - 算法的执行时间与输入实例的规模以及输入实例的构成相关。
- 编制不同的数据配置，分析算法的最好、最坏、平均工作情况
  - 最好运行时间：Bogus假象;
  - 最坏运行时间：任何输入实例的运行时间的上界;
  - 平均运行时间：需要对输入数据的分布做出假设。
- 一般考察算法的最坏运行时间
  - 对某些算法，最坏情况经常出现 (比如在数据库中搜索一个不存在的记录);
  - 对很多算法，平均情况往往与最坏情况大致一样;
  - 若平均时间和最坏时间不是同数量级，那么算法选择依据是：最好、最坏的概率较小时，尽量选择平均时间较小的算法。

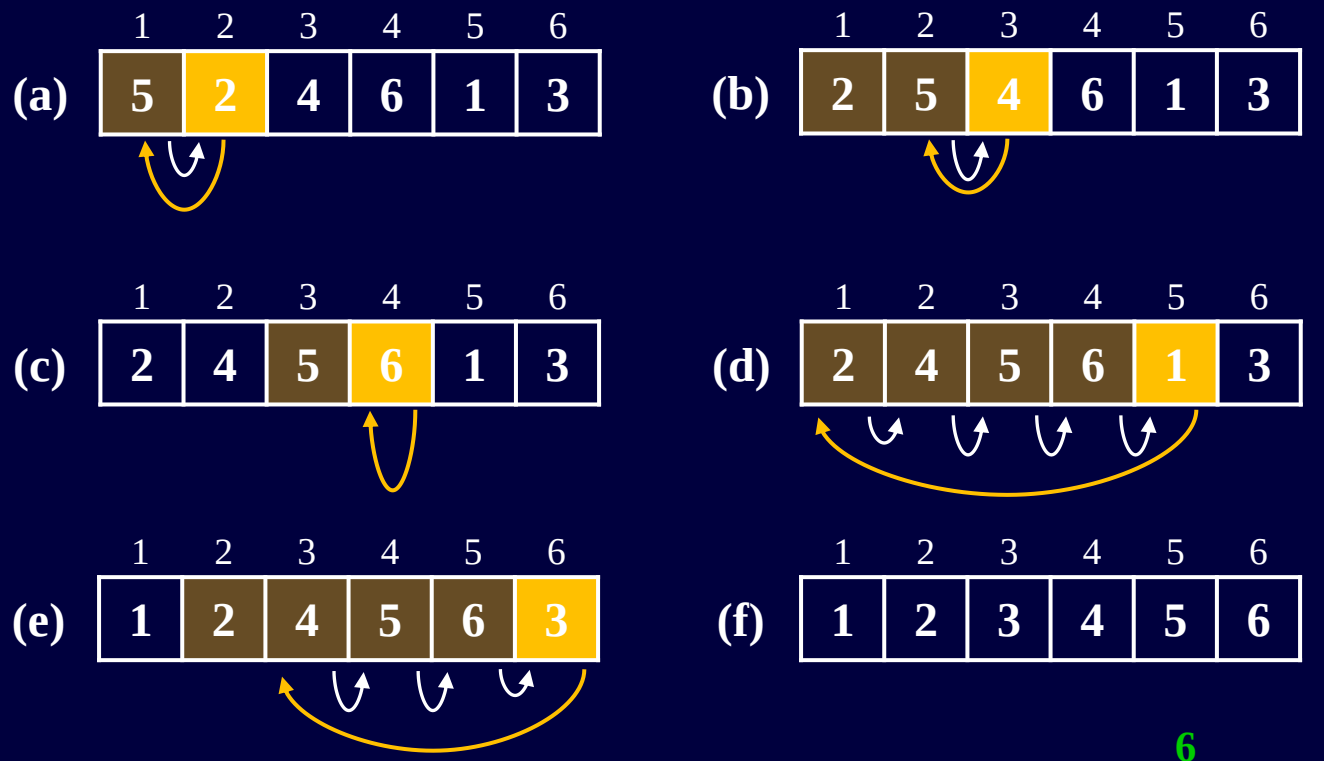
时间分析 (续)

- 算法运行时间
  - ①用基本操作的数目(执行步数)来度量，将任何基本操作看作花费单位时间，算法分析独立于机器；
  - ②用更接近实际的计算机上实现的时间来度量，不同的指令具有不同的执行时间，如RAM模型；
  - ①与②之间相差一个常数因子。

分析示例 —— 插入排序

- **排序问题：**
  - 输入(Input) ：$< a_1, a_2, \ldots, a_n >$
  - 输出(Output)：$< a'_1, a'_2, \ldots, a'_n >$
- **插入排序(Insertion Sorting)：**

```
INSERTION_SORT(A)
1  for j = 2 to A.length
2      key = A[j]
3      // Insert A[j] into the sorted sequence A[1..j − 1]
4      i = j − 1
5      while i > 0 and A[i] > key
6          A[i + 1] = A[i]
7          i = i − 1
8      A[i + 1] = key
```



分析示例 —— 插入排序 (续)

- **时间效率分析：**

$INSERTION\_SORT(A)$	cost	times
1 $for\ j = 2\ to\ A.length$	c_1	n
2 $key = A[j]$	c_2	$n-1$
3 // Insert $A[j]$ into the sorted sequence $A[1..j-1]$	0	$n-1$
4 $i = j-1$	c_4	$n-1$
5 $while\ i > 0\ and\ A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 $A[i+1] = A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7 $i = i-1$	c_7	$\sum_{j=2}^{n} (t_j - 1)$
8 $A[i+1] = key$	c_8	$n-1$

- **总运行时间：** $T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{j=2}^{n} t_j + c_6 \sum_{j=2}^{n} (t_j - 1) + c_7 \sum_{j=2}^{n} (t_j - 1) + c_8(n-1)$

分析示例 —— 插入排序 (续)

- 插入排序的运行时间分析：
  - ❖最好情况：初始数组有序

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5(n-1) + c_8(n-1)$$

$$= (c_1 + c_2 + c_4 + c_5 + c_8)n - (c_2 + c_4 + c_5 + c_8) \Rightarrow g(n) = n$$

(n的线性函数)

  - ❖最坏情况：初始数组反向有序

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5(n(n+1)/2 - 1) +$$

$$c_6(n(n-1)/2) + c_7(n(n-1)/2) + c_8(n-1)$$

$$= (c_5/2 + c_6/2 + c_7/2)n^2 + \Rightarrow g(n) = n^2$$

(n的二次函数)

$$(c_1 + c_2 + c_4 + c_5/2 - c_6/2 - c_7/2 + c_8)n - (c_2 + c_4 + c_5 + c_8)$$

算法分析方法

- 在分析插入排序的运行时间时，我们
  - 忽略精确运行时间中的**精确常量(exact constants)**；
  - 忽略精确运行时间中的**低阶项(low order terms)**。
- 我们关注**总运行时间**表达式中的**最高次项**，并**忽略常系数**
  - 当问题规模n足够大时，$T(n)$中的低阶项和常系数不如最高次项中的因子重要；
  - 当问题规模n较小时，$T(n)$中的低阶项和常系数的重要性不可忽略；
  - 需要对时间效率进行“**渐进分析**”。